Dynamical Behavior and Complexity of Langton's Ant

The Universality of the Ant

ANDRÉS MOREIRA, ANAHÍ GAJARDO, AND ERIC GOLES

Received October 20, 2000; revised March 27, 2001; accepted May 3, 2001

Langton's ant, with its intriguing behavior, has been elusive to theoretical results, particularly from the point of view of the system's complexity. We summarize here some recent work of our group, that sheds some new light on the ant.

One of the first models of artificial life, proposed back in the 1980s by Christopher Langton, founder of the field, was the *virtual ant* [1,2]. This simple cellular automaton is defined on the square grid in the following way: each square ("cell") of the grid can be in one of two states, white or black, and the ant is represented by a short arrow that stands on one cell and points to the north, the west, the east, or the south. At each time step, it moves to the cell it was pointing to, and it turns 90 degrees to the left if this cell is white or 90 degrees to the right if it is black; in addition, the state of the cell is switched. Figure 1a shows the situation after 5 time steps, starting with a background of only white cells. The interesting part starts when we let the ant go on with its walk. At iterations 96 and 184, rotational symmetry of order 2 is observed; at iteration 368 (Figure 1b), it is almost of order 4. When one could expect further symmetrical patterns, the symmetry breaks down, and after the step 500, the ant seems to walk at random, for more than 9000 iterations (Figure 1c). Again, when one would expect this chaos to go on forever, the ant suddenly starts building a pattern that is periodic but for a drift; it is the so called "highway," and in the absence of obstacles, the ant will draw it forever (Figure 1d).

This brief history involving a break of symmetry, a "chaotic" phase, and sudden order, all generated by a rule that could hardly be simpler, becomes more intriguing when we notice that it repeats with other initial patterns. In fact, the highway has appeared, sooner or later, in all the simulations that have been started with a finite amount of black (or of white) cells (we call them *configurations with finite support*). Nobody has demonstrated

Eric Goles is a mathematician, with two Ph.D.'s from the University of Grenoble; his research has focused on cellular automata. He currently presides the Chilean National Commission for Scientific and Technological Research. Anahí Gajardo, and Eric Goles are Ph.D., students under his direction at the Department of Mathematical Engineering and the Center for Mathematical Modeling at the Universidad de Chile, Casilla 170/30-Correo 3, Santiago de Chile. For further information on their work please see http://www.dim.uchile.cl/~agajardo/langton

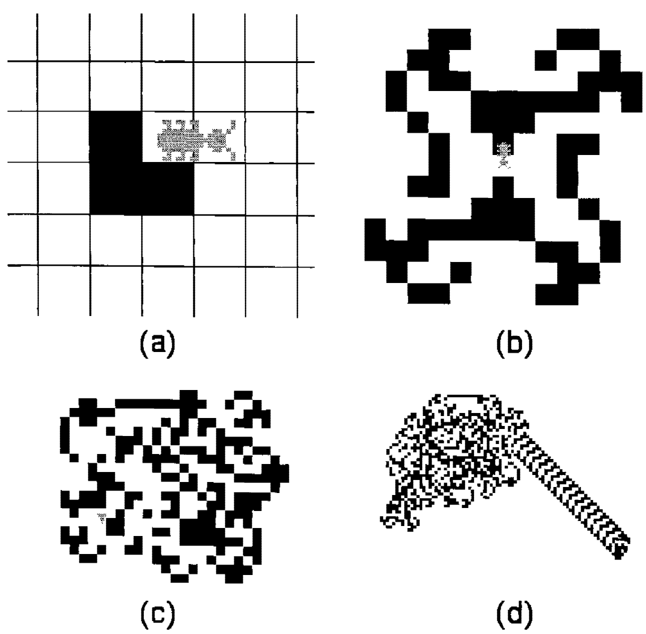
that this will always happen, nor has anyone found a counterexample; the lack of an answer for such an "easy" question about a simple deterministic system has even been cited as an example of the possible explanatory limitations of a "theory of everything" [3].

The ant is a rather natural system, as testified by at least two rediscoveries of its rule. The first time it was found by Greg Turk [4], a graduate student of computer sciences, who was experimenting with "two-dimensional" Turing machines. In a classical Turing machine, an infinitely long tape is divided into cells, each of which contains a symbol of a finite alphabet; a finite input written on the tape is read, one cell at each time, by a "head" that can be in any of a finite number of internal states. This internal state and the symbol found in the cell being read are used to look in a table (which characterizes the machine) and see what symbol must be written in the cell and whether the tape should be moved one cell forward or backward. Because there is no absolute position, one may also see the head as moving on a fixed tape; Turk saw it this way, and additionally, he used a two-dimensional "tape." This gave birth to a number of "tur-mites," moving around in the plane and drawing different patterns, and one of the simplest (with only one internal state, and two symbols in the tape's alphabet), which Turk noted to be interesting, was Langton's ant.

The ant was found again in the realm of physics, in the microscopic simulation of fluid dynamics. Various models have been used to study diffusion and percolation, all following the same general scheme: a particle moves on a lattice on which fixed scatterers are placed; when the particles collides with the scatterers, it modifies the direction of its movement.

The ant was found again in the realm of physics, in the microscopic simulation of fluid dynamics. Various models have been used to study diffusion and percolation, all following the same general scheme: a particle moves on a lattice on which fixed scatterers are placed; when the particles collide with the scatterers, it modifies the direction of its movement. In some models, the scatterers may be altered by the collisions. One of these models, proposed by Gunn and Ortuño [5], is similar to the ant, but with fixed scatterers. The Ruijgrok-Cohen Model [6] allows the particle to switch its orientation and is therefore equivalent to Langton's system (with the cells representing scatterers regularly placed in a two-dimensional space). The main exact result that has been obtained for the ant was demonstrated by Bunimovich and Troubetzkoy [7] and states that the trajectory of the ant on the square grid is always unbounded,

FIGURE 1



(a) Iteration 5, starting from a blank configuration; (b) Iteration 368, starting from a blank configuration; (c) Iteration 2000, starting from a blank configuration; (d) Iteration 10800, starting from a blank configuration.

regardless of the initial configuration. Wang [8] compared the dynamics of the ant on several grids and found bounded trajectories for the hexagonal case. Some other results can be found in Trobetzkoy [9] and Bunimovich [10], and several authors have performed empirical studies of the ant and some of its generalizations, including different underlying topologies, several ants, or more complicated rules. In the meantime, Langton's ant has become a popular subject of java applets, almost matching Conway's "Game of Life."

Despite all the efforts, there is still a lack of exact results; this was our reason to try some new approaches, which led to the results summarized here, results that can be read in detail in Gajardo [11] and Gajardo et al. [12]. The main one refers to the *universality* of the ant; it implies, in particular, the existence of undecidable problems. It is achieved through a construction of circuits that gives a lower bound for the computational complexity of the system. This and some additional results are explained in the following sections.

THE ANT PERFORMS LOGICAL CALCULATIONS

The main task was to perform logical calculations with the dynamics of the system. This is usually done through *Boolean circuits,* where a logical expression is represented as a one-way circuit, with the variables as inputs and the results as outputs. In parallel systems, the variables are usually represented through changing spots (say, gliders in Conway's Life). Because in our system all the dynamics takes place "under the feet" of the ant, the variables had to be represented through static spots, namely, through the states of some cells, which are "read" by the ant just by going over them and choosing its way accordingly. The path followed by the ant will be different depending on the values that the variables had (we used *white* to represent *false* or 0), and this will determine whether or not the ant passes over a certain cell that is predetermined to contain the result of the operation. In the initial configuration, this cell is white. If the ant does not visit it, then it will remain white, and we say that the answer was "false." If it does visit the cell, the state will be switched, and we read "true."

To see how it works, consider the simplified scheme in Figure 2a, which represents a circuit that takes two inputs and repeats them as outputs, exchanging their values. The ant will visit Output 2 if and only if it turns to the right at Input 1 (i.e., the cell is black), so that the resulting state in Output 2 will be the same that was initially contained in Input 1; the same applies for Input 2 and Output 1.

To embed this "logical gate" in the square grid, we required to draw "paths" (which are usually straight lines of black states leading the ant in a desired direction) and to design some special configurations to cross and join these paths. The result is shown in Figure 5b.

By joining the paths, we force the ant to always exit the configuration through the same cells, on the right (having entered on the left). Furthermore, the (possible) visit of the ant to the Output cells is done "from above," thus allowing their use as Input cells for other gates. These features allow us to combine different gates in successive rows: the ant will go through a row of gates, calculating the outputs, and then it will go to another row below the first one and do the same, and so forth. We built several gates, all of them similar to the one of Figure 2b, that can copy a variable, switch it, duplicate it, or perform the AND operation between two variables. Any Boolean circuit may be written in terms of these gates, and it is therefore possible, for any logical expression, to build an initial configuration of the ant system in which a few cells at the top receive the values of the variables, and the result of the calculation is found at the bottom, after a walk of the ant.

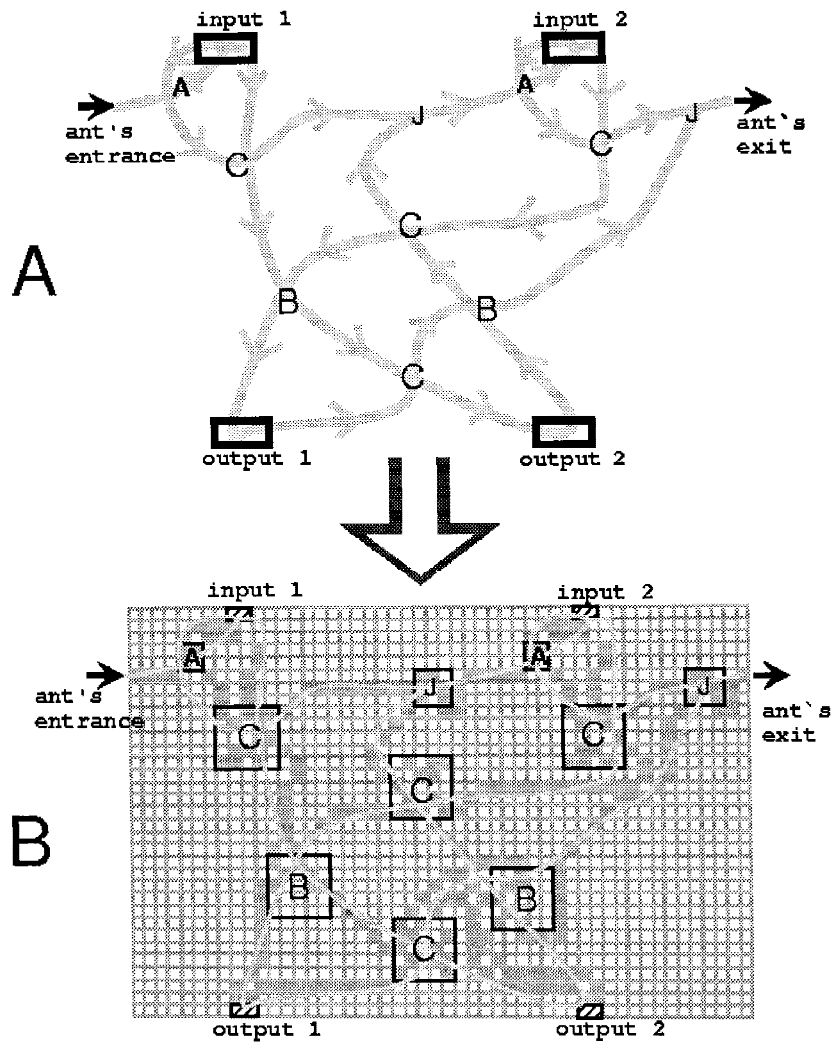
Let us consider now the consequences of this construction. The problem of calculating the output of Boolean circuits is known to be in class *P*: this means, there is an algorithm that takes a circuit and a given assignment of variable values and gives the answer after a time that is polynomially bounded in the size of its inputs. Moreover, the problem is known to be *P-complete,* which means that any problem in *P* can be reduced to it. If a problem has this last feature, without been itself known to be in *P,* it is said to be *P-hard*. Thus, to show that a problem is *P-hard,* we just have to reduce a *P-complete* problem to it (with some technical restrictions on the reduction) and that is exactly what we did: the problem of Boolean circuit calculation was reduced to the problem of answering: "For this given initial configuration, does the ant visit this given cell before this other given cell?" (Consider the last output cell, and the cell used to exit the configuration). The prediction of the ant's trajectory is then a *P-hard* problem, and so we get a first lower bound for the computational complexity of the system.

For a given finite circuit, the construction described above is also finite. Hence, the *P-hard* problem described above may be also associated with the graph that consists of finite blocks of the grid. We built some examples of such graphs, in which the length of the ant's trajectory grows exponentially as the number of cells increases; if the only general way of predicting the trajectory is the explicit simulation (which is likely to be the case), then this exponential growth gives an idea of its high unpredictability.

UNIVERSALITY

A *cellular automaton* (CA) is a dynamical network of cells, where each cell updates its internal state at each time step as a function of the previous values of itself and its neighbors. In one dimension, the cells form a line. A CA can have a *quiescent* state: if a cell and its neighbors are in it, then the state of the cell will be conserved at the next iteration. Thus,

FIGURE 2

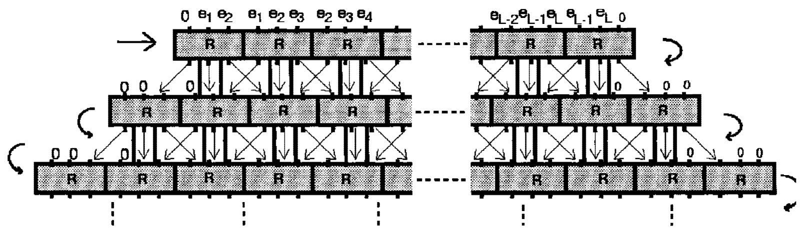


A gate to cross two logical variables: scheme, and construction on the square grid. (*To-left* states are painted gray in B).

all the dynamics of the system takes place at the nonquiescent cells and their neighbors. If the system starts with a configuration where all but a finite number of cells are in a quiescent state (what we call a *finite* configuration), then this finiteness will be conserved: at each time step, we only need to update a finite number of cells, in a restricted—but possibly growing—region. For any CA, we can represent its states with binary numbers and use a Boolean circuit to calculate the output of the local transition function. In Figure 6 we show how the repetitions of that circuit ("R") can be arranged to allow the ant to compute the successive iterations of the CA, if the CA has a quiescent state and the initial configuration is finite. By writing a starting configuration and letting the ant walk, we obtain the complete space-time diagram of the CA.

Turing machines, described in the Introduction, are the formalization of algorithms; any computer program can be encoded as a Turing machine. A *universal Turing machine* (UTM) is one that takes as inputs the codification of any Turing machine and the contents of a tape and simulates the dynamics that the encoded machine would have if it runs on that tape; it is a universal computer. Any Turing machine can be simulated by a one-dimensional cellular automaton, which happens to have a quiescent state. Thus, the walk of the ant is able to simulate a UTM, if the appropriate initial configuration is provided. If we add a mechanism of stop marks, the trapezoid of Figure 6 can be replaced by a periodic tiling, and so the initial configuration for the UTM will be periodic, but for a finite region on which the input is written.

FIGURE 3

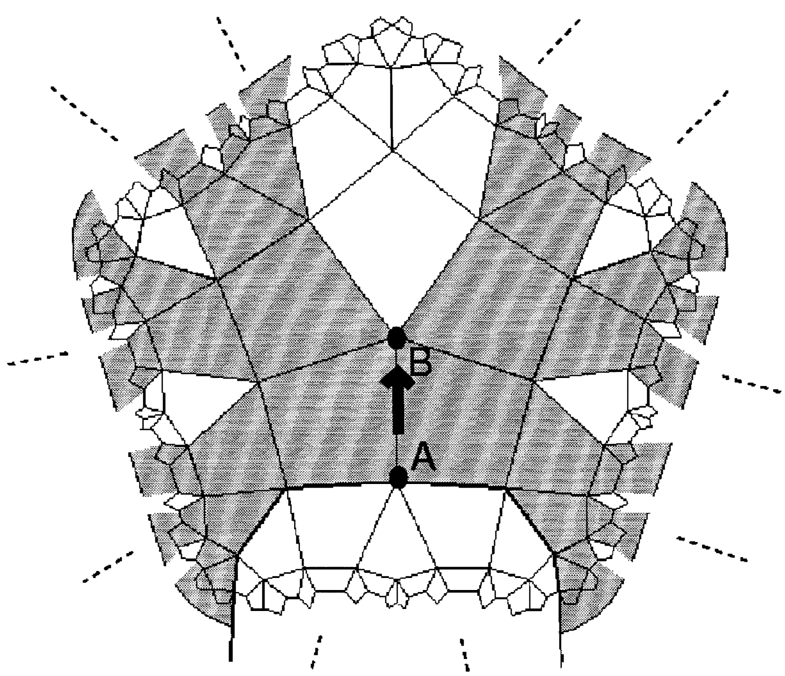


The ant simulates each iteration of the CA in a row of gates, crosses the repetitions of the outputs (preparing the next input), and goes to the next row. R, the circuit that calculates the rule.

As a by-product, the simulation of CA (or the universality of the system) implies the existence of undecidable questions. It is known that no algorithm can exist that takes a CA and its initial configuration as inputs, and answers in a finite time whether or not a given partial configuration will ever appear in the corresponding space-time diagram. The translation to the ant system is the following: there is no algorithm that takes an initial configuration C of the grid, and a given block (partial configuration), and predicts if the later will ever appear on the grid, if the ant starts walking on C. Notice that this statement applies to the class of initial configurations that are periodic up to some finite alteration; if we restrict the class to the finite configurations, then the question may be decidable (and it will be, if the conjecture about the highway happens to be true).

The usual grids on which the ant has been studied are the square, the hexagonal, and the triangular ones. They belong to the larger family of biregular graphs $\Gamma(k,d)$, where all cells have k neighbors and each "face"—the smallest area surrounded by edges—has d adjacent faces. The three grids correspond to $\Gamma(4,4)$, $\Gamma(3,6)$ and $\Gamma(6,3)$, respectively, and are the only infinite Euclidean graphs of the family; the rest are

FIGURE 4



For highly connected planar graphs, the trajectory of the ant is eventually restricted to a zone like the one shadowed here, relative to an ant going from A to B. Notice that here the nodes (the intersections of the lines) take the place of the cells of the square lattice.

either finite—a few cases—or hyperbolic. The study of the general family is not only a way to include different geometries but also a help to understand what topological features are important for the constructions and theorems; it is also a way to approximate general, arbitrary graphs.

A general form for the construction of gates and circuits was found, which applies to all $\Gamma(k,d)$ with $d = 3,4$, and implies the universality of the ant on all of them (including the hexagonal grid). When $d \geq 5$, the construction of circuits could not be done. Indeed, for a class of graphs that includes all the infinite $\Gamma(k,d)$ with $d \geq 5$, there appear strong restrictions on the dynamics: for every position of the ant, there is a large portion of the graph that cannot be reached, no matter what the states in the cells are; the trajectories are limited to a tree-shaped zone, as shown in Figure 7 (this result generalizes the one proved in Kong and Cohen [13] for the triangular grid). For finite initial configurations, the ant eventually starts building a kind of highway (what is only conjectured for the square grid), and in general, the system seems to be far less complex.

CONCLUSIONS

A first interesting observation that is derived from our work is the difference between the regular graphs of degree 5 or more and the ones of degrees 3 and 4. The presence of a blocking mechanism in the highly connected graphs restricts the dynamics and the memory of the ant, making our construction of circuits impossible; we conjecture that no kind of universality will be found in these cases.

For lower degrees, on the other hand, and particularly for the square and the hexagonal grids, we find a high unpredictability. The existence of *P*-hard problems associated to the dynamics gives a lower bound for the *computational complexity* of the system (the amount of work needed to predict its behavior).

The existence of undecidable problems, that is, the impossibility of having general algorithms to answer some questions in finite time, even when only configurations described by a finite amount of information are considered, is a further hint on the complexity. It is a noticeable point, given the simplicity and determinism of the law that rules the ant's universe.

The main result is the universality of the system, which refers to the *computational power* of its dynamics and adds the ant to the list of surprisingly simple systems that are able to run any algorithm and do anything a computer can do; some other examples are Conway's Game of Life [14] and sand piles [15]. Universality means, for instance, that the ant can spend the eternity writing on the grid the consecutive digits of Pi; if it starts with the appropriate periodic background and the appropriate finite perturbation; it could also list the consecutive prime numbers, and so forth. Some years ago it was called "the industrious ant" [2], just because of its laborious travels, without knowing if it could produce something more than a mess. Now we know it can.

The main result is the universality of the system, which refers to the *computational power* of its dynamics and adds the ant to the list of surprisingly simple systems that are able to run any algorithm and do anything a computer can do....

REFERENCES

1. C. Langton. Physica D 10: 1986, pp. 120–149.
2. D. Gale. Tracking the automatic ant and other mathematical explorations. Springer-Verlag, New York, 1998.
3. I. Stewart. Scientific American July: 1994, pp. 104–107.
4. A. K. Dewdney. Scientific American September: 1989, pp. 124–127.
5. J. Gunn, M. Ortuño. Journal of Physics A 18: 1985, L1035–L1101.
6. Th.W. Ruijgrok, E. Cohen. Physics Letters A 133: 1988, pp. 415–418.
7. L. Bunimovich, S. Troubetzkoy. Journal of Statistical Physics 67: 1992, pp. 289–302.
8. F. Wang. Ph.D. Dissertation. The Rockefeller University, 1995.
9. S. Troubetzkoy. Alabama Journal of Mathematics 21: 1997, pp. 3–13.
10. L. Bunimovich. International Journal of Bifurcation and Chaos 6(N.6): 1996, pp. 1127–1135.
11. A. Gajardo, A. Moreira, E. Goles. Discrete Applied Mathematics, to appear (2001).
12. A. Gajardo, E. Goles, A. Moreira. STACS 2001. In: Lecture Notes in Computer Science N. 2010. A. Ferreira and H. Reichel (Eds.), Springer-Verlag, Berlin 2001, pp. 259–270.
13. X. Kong, E. Cohen. Journal of Statistical Physics 62: 1991, pp. 737–757.
14. E. Berlekamp, J. Conway, R. Guy. Winning ways for your mathematical plays, Vol. 2, New York, Academic Press, 1982.
15. E. Goles, M. Margenstern. International Journal of Modern Physics C 7(N.2): 1996, pp. 113–122.